Amendments to the Claims

Applicant respectfully requests reconsideration of this application as amended. Claims 1-4, 6, 8, 10-13, 37, and 42-50 have been amended. Claims 5, 7, and 14 have been canceled without prejudice. No claims have been added.

Listing of Claims:

1. (Currently Amended) A method of parallel processing bit-synchronous High-level Data Link Control [[HDLC]] data, comprising:

storing a current at least two bytes of bit-synchronous High-level Data Link Control [[HDLC]] data in a shift register, wherein, at each successive clock cycle, a new incoming byte is shifted into said shift register and an old byte is shifted out of said shift register;

processing in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register so as to detect a start-of-frame [[SOF]] sequence within said current at least two bytes in said shift register during a first clock cycle with a plurality of comparators coupled with said shift register, wherein each comparator is coupled with said shift register to parallel process a unique combination of eight successive bits within said shift register;

processing in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register so as to detect an end-of frame[[EOF]] sequence within said current at least two bytes in said shift register during at least one subsequent clock cycle with said plurality of comparators coupled with said shift register; and

sending valid payload data bits to a packer logic unit, wherein said valid payload data bits comprise at least some bits shifted into said shift register between said start-of-

frame [[SOF]] sequence and said end-of frame[[EOF]] sequence.

2. (Currently Amended) The method of claim 1 further comprising processing in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register to detect an Abort sequence during said at least one subsequent clock cycle, wherein if said Abort sequence is detected, all bits received after said start-of-frame [[SOF]] sequence are discarded and a search for a new start-of-frame [[SOF]] sequence is initiated.

3. (Currently Amended) The method of claim 1 further comprising:

processing in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register to detect at least one stuff bit; and

discarding said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, excluding said at least one stuff bit.

4. (Currently Amended) The method of claim 1 further comprising discarding all bits received between said start-of-frame [[SOF]] and end-of frame[[EOF]] sequences if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

5. (Canceled)

6. (Currently Amended) A method of parallel processing bit-synchronous data, comprising:

storing a current at least two bytes of bit-synchronous data in a shift register, wherein a newly received byte is shifted in and an old byte is shifted out of said shift register at each clock cycle;

processing in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register so as to detect valid payload data bits within said current at least two bytes using a plurality of comparators coupled with said shift register to search for a specified sequence of bits stored within said at least two bytes during a first clock cycle, wherein each comparator is coupled with said shift register so as to parallel process a unique combination of eight successive bits contained within said shift register; and

storing detected valid payload data bits in a packer logic unit for further processing.

7. (Canceled)

8. (Currently Amended) The method of claim [[7]] 6 wherein said act of searching for a specified sequence comprises searching for a start sequence within said shift register, wherein said valid payload data bits comprise at least some bits received after said start sequence.

9. (Original) The method of claim 8 further comprising searching for an end sequence within said shift register, after said start sequence has been detected, during at least one clock cycle subsequent to said first clock cycle, wherein said valid payload data bits comprise at least some bits received between said start and end sequences.

10. (Currently Amended) The method of claim 9 wherein said bit-synchronous data comprises bit-synchronous High-level Data Link Control [[HDLC]] data, said start sequence comprises a start-of-frame [[SOF]] sequence and said end sequence comprises an end-of frame[[EOF]] sequence.

11. (Currently Amended) The method of claim 10 further comprising processing in parallel ~~a plurality of bits~~ said current at least two bytes stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said start-of-frame [[SOF]] sequence are discarded and a search for a new start-of-frame [[SOF]] sequence is initiated.

12. (Currently Amended) The method of claim 10 further comprising:

processing in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register to detect at least one stuff bit; and

discarding said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, excluding said at least one stuff bit.

13. (Currently Amended) The method of claim 10 further comprising discarding all bits received between said start-of-frame [[SOF]] and end-of frame[[EOF]] sequences if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

14. (Canceled)

15. (Currently Amended) The method of claim 6 further comprising de-scrambling said current at least two bytes prior to storing said current at least two bytes in said shift register, wherein said de-scrambling comprises de-scrambling at least eight bits in parallel during a single clock cycle.

16.-36. (Canceled)

37. (Currently Amended) A system for parallel processing bit-synchronous data, comprising:

a shift register ~~for storing~~ to store a plurality of bits of bit-synchronous data that comprises bit-synchronous High-level Data Link Control [[HDLC]] data, said shift register stores a current at least two bytes of said bit-synchronous High-level Data Link Control [[HDLC]] data, wherein a new plurality of bits is shifted in and an old plurality of bits is shifted out of said shift register during successive clock cycles, said new and old plurality of bits each comprise one byte of data;

a de-framer unit, coupled to said shift register, ~~for detecting~~ to detect valid payload data, wherein said de-framer unit processes in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register during ~~a first~~ said clock cycles and searches for a start sequence comprised of a start-of-frame [[SOF]] sequence and an end sequence within said shift register, said de-framer unit comprises a plurality of comparators wherein each comparator is coupled to a unique subset of eight successive bits contained within said shift register, and said valid payload data bits comprise at least

some bits received between said start-of-frame [[SOF]] and end-of frame[[EOF]] sequences; and

a packer logic unit, coupled to said de-framer unit, ~~for storing~~ to store said valid payload data received from said de-framer unit, said de-framer unit further comprising a look-up table (LUT) representing a plurality of bit sequences containing at least one stuff bit and wherein, if said at least one stuff bit is detected, said LUT passes through valid payload data bits to said packer logic unit while not allowing said at least one stuff bit to be sent to said packer logic unit.

38. (Previously Presented) The system of claim 37 wherein said packer logic unit discards said valid payload data received from said de-framer unit if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

39. (Previously Presented) The system of claim 37 wherein said de-framing unit further processes in parallel said ~~a plurality of bits~~ current at least two bytes within said shift register to detect a specified sequence of bits during said first clock cycle.

40. (Previously Presented) The system of claim 39 wherein said de-framing unit searches for a start sequence within said shift register, wherein said valid payload data bits comprise at least some bits received after said start sequence.

41. (Previously Presented) The system of claim 40 wherein said de-framing unit further searches for an end sequence within said shift register, after said start sequence has been detected, during at least one subsequent clock cycle ~~subsequent to said first~~

~~clock cycle~~, wherein said valid payload data bits comprise at least some bits received between said start and end sequences.

42. (Currently Amended) The system of claim 37 wherein said de-framer unit further processes in parallel ~~a plurality of bits~~ said current at least two bytes stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said start-of-frame [[SOF]] sequence are discarded and a search for a new start-of-frame [[SOF]] sequence is initiated.

43. (Currently Amended) The system of claim 37 wherein said de-framer unit further processes in parallel ~~a plurality of bits~~ said current at least two bytes within said shift register to detect at least one stuff bit and discards said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, excluding said at least one stuff bit.

44. (Currently Amended) A bit-synchronous High-level Data Link Control [[HDLC]] engine, comprising:

a shift register ~~for storing~~ to store at least two bytes of bit-synchronous High-level Data Link Control [[HDLC]] data, wherein a new byte is shifted in and an old byte is shifted out of said shift register during each successive clock cycle; and

a de-framer unit, coupled to said shift register, ~~for detecting~~ to detect valid payload data within said shift register, said de-framer unit to detect[[s]] a start-of-frame

[[SOF]] sequence within said at least two bytes during [[said]] a first clock cycle and to detect[[s]] an end-of frame[[EOF]] sequence within said at least two bytes during a subsequent clock cycle, wherein said valid payload data comprises at least some of the bits received between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, said de-framer unit comprises a plurality of comparators ~~for detecting~~ to detect a specified sequence of bits within said at least two bytes in said shift register during [[a]] said [[first]] clock cycles, wherein each comparator is coupled to a unique subset of eight successive bits contained within said shift register, and said de-framer unit further processes in parallel a plurality of bits within said shift register to detect at least one stuff bit and discards said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, excluding said at least one stuff bit; and

a packer logic unit, coupled to said de-framer unit, ~~for storing~~ to store said valid payload data received from said de-framer unit, wherein said de-framer unit further comprises a look-up table (LUT) representing a plurality of bit sequences containing at least one stuff bit, wherein, if a stuff bit is detected, said LUT passes through valid payload data bits while not allowing said at least one stuff bit to be sent to said packer logic unit.

45. (Currently Amended) The bit-synchronous High-level Data Link Control [[HDLC]] engine of claim 44 further comprising a packer logic unit, coupled to said de-framer unit, ~~for storing~~ to store said valid payload data received from said de-framer unit, wherein said packer logic unit ~~discards~~ to discard said valid payload data received from

said de-framer unit if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

46. (Currently Amended) The bit-synchronous High-level Data Link Control [[HDLC]] engine of claim 44 wherein said de-framer unit to further process[[es]] in parallel a plurality of bits stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said start-of-frame [[SOF]] sequence are discarded and a search for a new start-of-frame [[SOF]] sequence is initiated.

47. (Currently Amended) A bit-synchronous High-level Data Link Control [[HDLC]] engine, comprising:

a shift register ~~for storing~~ to store at least two bytes of bit-synchronous High-level Data Link Control [[HDLC]] data, wherein a new byte is shifted in and an old byte is shifted out of said shift register during each successive clock cycle;

a de-framer unit, coupled to said shift register, ~~for detecting~~ to detect valid payload data within said at least two bytes in said shift register, said de-framer unit detects a start-of-frame [[SOF]] sequence within said at least two bytes during [[said]] a first clock cycle and ~~detects~~ to detect an end-of frame[[EOF]] sequence within said at least two bytes during a subsequent clock cycle, said valid payload data comprises at least some of the bits received between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, and said de-framer unit comprises a plurality of comparators ~~for detecting~~ to detect a specified sequence of bits within said shift register during [[a]] said [[first]] clock cycles, wherein each comparator is coupled to a unique subset of eight successive bits contained within said shift register; and

a packer logic unit, coupled to said de-framer unit, ~~for storing~~ to store said valid payload data received from said de-framer unit, wherein said packer logic unit discards said valid payload data received from said de-framer unit if an amount of valid payload data bits received constitute less than a minimum number (N) of bytes.

48. (Currently Amended) The bit-synchronous High-level Data Link Control [[HDLC]] engine of claim 47 wherein said de-framer unit to further process[[es]] in parallel ~~a plurality of bits~~ said at least two bytes stored in said shift register to detect an Abort sequence during at least one clock cycle subsequent to said first clock cycle, wherein if said Abort sequence is detected, all bits received after said start-of-frame [[SOF]] sequence are discarded and a search for a new start-of-frame [[SOF]] sequence is initiated.

49. (Currently Amended) The bit-synchronous High-level Data Link Control [[HDLC]] engine of claim 47 wherein said de-framer unit to further process[[es]] in parallel ~~a plurality of bits~~ said at least two bytes within said shift register to detect at least one stuff bit and discards said at least one stuff bit, if detected, wherein said valid payload data bits comprise all bits shifted into said shift register between said start-of-frame [[SOF]] sequence and said end-of frame[[EOF]] sequence, excluding said at least one stuff bit.

50. (Currently Amended) The bit-synchronous High-level Data Link Control [[HDLC]] engine of claim 49 wherein said de-framer unit further comprises a look-up table (LUT) representing a plurality of bit sequences containing at least one stuff bit,

wherein, if a stuff bit is detected, said LUT passes through valid payload data bits while not allowing said at least one stuff bit to be sent to said packer logic unit.